A Survey on Software Fault Prediction Technique based on Clustering Algorithm and Artificial Intelligence

Margi Patel
Computer Engineering,
Silver Oak College of Engineering & Technology, Ahmedabad, Gujarat, India

***Abstract* – There has been rapid growth of software development. During Transmission of Data faults are Created. However software Fault Prediction Techniques are used to Detect Fault. Software Fault Prediction improve the quality and reliability of software by predicting faults .Quality of Software measure in term of fault proneness of data .These software defect may lead to degradation of the quality which might be the cause of failure. In this paper focus on clustering with large dataset and predicting faults efficiently. We show a comparatively analysis of software fault prediction based on clustering technique, neural network method, statistical method. Fault prediction reduce the overall time and less data processing.**

***Index Terms*- Fault Prediction, Clustering Technique, Neural Network Technique, Statistical method**

I. INTRODUCTION

SOFTWARE quality and reliability are main concerns in modern era.It is widely accepted that software with defects lacks quality.Real time software application and complex software systems demands high quality.A software system contain many modules and any of these can contain faults[5].

Clustering is a division of data into groups of similar objects. Each group called cluster consists of objects that are similar between themselves and dissimilar to objects of other groups. Clustering is a method of unsupervised learning, and a common technique for statistical data analysis used in many fields, including machine learning, data mining, pattern recognition, image analysis and bioinformatics. Classification and prediction that can be used to extract models describing significant defect data classes or to predict future defect trends. Classification predicts categorical or discrete, and unordered labels, whereas prediction models predict continuous valued functions. Such analysis can help us for providing better understanding of the software defect data at large.

The underlying software engineering assumption is that the faultprone software modules will have similar software measurements, and hence are likely to be grouped together in the same cluster(s). Similarly, the not fault-prone modules will likely be grouped in the same cluster[2]. Clustering is an approach that uses software measurement data consisting of limited or no fault-proneness data for analyzing software quality[3].Clustering algorithms are being successfully applied for solving both classification and regression problems. It is therefore important to investigate the capabilities of this algorithm in predicting software quality[10].

Early prediction of software fault at coding phase can result in decrease cost and effort for software development. So, it is better to categorize the software module in faulty / non -faulty module just after completing the coding phase. module contain error derived as fault prone module. Software fault prediction uses historical and development data to identify fault in software. Various techniques have been applied for software fault prediction like partional clustering, hierarchical clustering, neural network, naive bayes, support vector machine and many more.

A software fault is a defect that causes software failure in an executable product. In software engineering, the nonconformance of software to its requirements is commonly called a bug. Software Engineers distinguish between software faults, software failures and software bugs. In case of a failure, the software does not do what the user expects but on the other hand fault is a hidden programming error that may or may not actually manifest as a failure and the non-conformance of software to its requirements is commonly called a bug[3].

II. PROBLEM DEFINITION

The problem under investigation is in the area of software measurement and fault prediction. More precisely, the study focuses on investigation of an adaptive fault prediction approach that exploits existing prediction techniques, adapting them to improve their ability to predict faulty system modules across different software projects. we will cover the popular model of defect prediction and evolute the pros and cons of each model. To improvise the speed of processing and increase accuracy in transmitted data stream at receiver side.`

III. STUDY OF SOFTWARE FAULT PREDICTION MODELS

There are several different technique have been proposed to develope predictive software metrics for the classification of software module into fault-prone and non fault-prone categories.

3.1 K-means Clustering

One of the simplest clustering algorithms is K-means clustering method. The k-means algorithm assigns each point to the cluster whose center (also called centroid) is nearest. The center is the average of all the points in the cluster — that is, its coordinates are the arithmetic mean for each dimension separately over all the points in the cluster[2].

Most of time, K-Means computationally faster than hierarchical clustering and k-Means produce tighter clusters than hierarchical clustering, especially if the clusters are globular.K-Means is applied in data compression, data modelling, expression analysis and other fields[5].

3.2 Fuzzy C-means Clustering

Fuzzy c-means clustering method was developed by Bezdek. Each instance can belong to every cluster with a different membership grades between 0 and 1 for this algorithm . A dissimilarity function, as is minimized and centroids which minimize this function are identified[2]. Fuzzy C Means clustering algorithm is being used for predictive models to predict faulty/non faulty modules. Fuzzy c-means (FCM) is a method of clustering which allows one piece of data to belong to two or more clusters[4]. The fuzzy *c*-means algorithm is very similar to the *k*-means algorithm.

3.3 Hierarchical Clustering

Hierarchical clustering creates a hierarchy of clusters which may be represented in a tree structure called a dendrogram.The root of the tree consists of a single cluster containing allobservations, and the leaves correspond to individual observations. They are either agglomerative (bottom-up) or divisive (top-down). Hierarchical Clustering algorithm for clustering of the software components into faulty/fault-free systems. Clustering can be a very effective technique to identify natural groupings in data from a large data set, thereby allowing concise representation of relationships embedded in the data. In our study, clustering allows us to group software modules into faulty and non-faulty categories hence allowing for easier understandability. There are two main methods of hierarchical clustering algorithm. First method is agglomerative approach, where we start from the bottom where all the objects are and going up (bottom up approach) through merging of objects. We begin with each individual objects and merge the two closest objects. The process is iterated until all objects are aggregated into a single group. Second method is divisive approach (top down approach), where we start with assumption that all objects are group into a single group and then we split the group into two recursively until each group consists of a single object. One possible way to perform divisive approach is to first form a minimum spanning tree (e.g. using Kruskal algorithm) and then recursively (or iteratively) split the tree by the largest distance[10].

3.4 Neural Network

An Artificial Neural Network (ANN) or simply a Neural Network (NN) is an information-processing paradigm that is inspired by the way a biological nervous system in human brain works. An artificial neuron is a small processing unit and performs a simple computation that is fundam to the operation of a neural network. The model of a neuron contains the basic elements like inputs, synaptic weights and bias, summing junction and activation function. ANN can be divided into two major categories based on their connection topology: Feed forward and Feed backward neural networks. Feed-forward neural networksallow the signal to flow in the forward direction only. The signals from any neuron do not flow to any other neuron in the preceding layer. In Feed backward neural networks the signal from a neuron in a layer can flow to any other neuron whether it be preceding or succeeding layers[18].

Multi Layer Perceptron

MLP is a fully connected feed forward artificial network model. It requires a desired output and is therefore term as supervised network . It organizes thin neuron into 3 layers .The initial layer is called as input layer, the intermediate layers are called as Hidden layers and the last layer is output layer. The classifier aims at creating a model, which correctly maps input to output using historical data, and therefore it help in prediction of number of defects in a class .morever the model used is simple enough, as simple MLP model are more robust in nature and also helps reduce complexity of time[1].

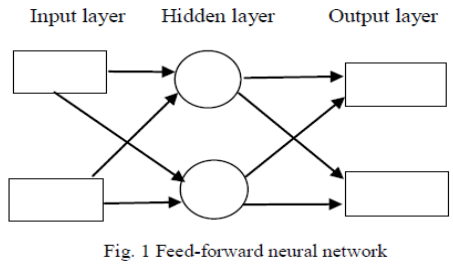

Fig. 1 Feed-forward neural network

Fig 1: Feed Forward Neural network [5]

3.5 Bayesian Network

Bayesian classifier are also called as Probabilistic White-boxClassifier calculates later possibility allocation of any division of complex variable, while variables in the complementary devision are observed. This makes BN to act as Statistical classifier[1].

3.6 Naïve Bayes

NB is powerful and productive classifier and most widely used in the area of SDP as it performs binary classification.it calculates the possibility of each class of vector with experimental inputs of each system division using bayes rule.it is considered to provide better accuracy in comparision with other classifiers . NB provides computational efficiency and is easy to construct, as no learning phase is required[1].

IV. RELATED WORKS

Dhankhar, Swati, Himani Rastogi, and Misha Kakkar Proposed Software Fault Prediction Performance using Bayesian network,Naïve bayes,Neural Network.By using this method improve software quality and testing efficiency by early identification of fault.Neural Network classification model are more superior to other network model[1].

Gupta, Deepika, Vivek K. Goyal, and Harish Mittal Proposed Estimating of Software Quality with Clustering Techniques. this paper focus on clustering with very largedataset and very many attribute of different types.effective result can be produced by using fuzzy c-mean clustering[2] .

Kaur, Arashdeep, Parvinder S. Sandhu, and Amanpreet Singh Bra proposed Early software fault prediction using real time defect data.Predicting fault early in software life cycle can be used to improve software process control and achieve high software reliability.best prediction model is fusion of requirement and code metric model[3].

Kaur, Arashdeep, Amanpreet Singh Brar, and Parvinder S. Sandhu Proposed An empirical approach for software fault prediction. In this paper we investigate that wether the metrics available in the early lifecycle can be used to predict fault prone area or not. fuzzy c mean is better than k-mean in in case of requirement and combination metric model[4].

Shyna Kakkar, Amanpreet Singh Dhanoa Proposed Software Fault Prediction using hybrid k-mean feed forward neural network.this paper used hybrid approach to predict faults in software system .k-mean feed forward neural network has better acuuracy than fuzzy cmeans feed forward neural network. It can help in directing testeffort,reducing cost,increase quality of software and its reliability[5]

Now we proceed towards methodology and terminology of some required terms.

To predict the results, we have used confusion matrix. The confusion matrix has four categories: True positives (TP) are the modules correctly classified as faulty modules. False positives (FP) refer to fault-free modules incorrectly labeled as faulty. True negatives (TN) are the fault-free modules correctly labeled as such. False negatives (FN) refer to faulty modules incorrectly classified as fault-free modules.

Table 1 A confusion matrix of prediction outcomes

		Real data	
		Fault	No Fault
Predicted	Fault	TP	FP
	No Fault	FN	TN

Fig 2: Confusion Matrix[4]

The following set of evaluation measures are being used to find the results[4].

a). Probability of Detection (PD): it also called recall or specificity, is defined as the probability of correct classification of a module that contains a fault.

$$PD = TP / (TP + FN)$$

b). Probability of False Alarms (PF): PF is defined as the ratio of false positives to all non defect modules.

$$PF = FP / (FP + TN)$$

Basically, PD should be maximum and PF should be minimum. PD defines exact identification of faults whereas PF gives the cost to validate that faults[4].

C) Accuracy: It indicates proximity of measurement results to the true value, precision to the repeatability or reproducibility of the measurement. The accuracy is the proportion of true results (both true positives and true negatives) in the population. As represented in equation below[10

$$Accuracy = (TP+TN)/ (TP+FP + TN+FN)$$

d) Mean Absolute Error (MAE) : Mean absolute error, MAE is the average of the difference between predicted and actual value in all test cases; it is the average prediction error [5].

e) Root Mean Squared Error: RMSE is simply square root of mean squared error.the root mean squared error gives the error value as the same dimensionality as the actual and predicted value.

MAE and RMSE should always be less for better prediction. Accuracy, MAE and RMSE values are measured to evaluate the performance.

V. CONCLUSION

Based on survey, A variety of software fault prediction technique have been proposed,but none has proven to be consistently accurate. these technique include clustering technique, statistical method ,machine learning method, neural network method. Most of the researches were carried out with the help of NASA defect dataset. we would like to express NASA MDP organization for making their defect datasets publicly available. hybrid method is best for modelling fault proneness prediction in software system.

REFERENCE

[1] Dhankhar, Swati, Himani Rastogi, and Misha Kakkar. "SOFTWARE FAULT PREDICTION PERFORMANCEIN SOFTWARE ENGINEERING.",IEEE-2015.

[2] Gupta, Deepika, Vivek K. Goyal, and Harish Mittal. "Estimating of Software Quality with Clustering Techniques." Advanced Computing and Communication Technologies (ACCT), 2013 Third International Conference on. IEEE, 2013.

[3] Kaur, Arashdeep, Parvinder S. Sandhu, and Amanpreet Singh Bra. "Early software fault prediction using real time defect data." Machine Vision, 2009. ICMV'09. Second International Conference on. IEEE, 2009.

[4] Kaur, Arashdeep, Amanpreet Singh Brar, and Parvinder S. Sandhu. "An empirical approach for software fault prediction." Industrial and Information Systems (ICIIS), 2010 International Conference on. IEEE, 2010.

[5] Shyna Kakkar, Amanpreet Singh Dhanoa." Software Fault Prediction using hybrid k-mean feed forward neural network" IJCSEE-2015.

[6] PSO Optimized Software Fault Prediction system using Fuzzy C –Means,IJDACR- Volume 3, Issue 6, January 2015.

[7] Soleimani, A., and F. Asdaghi. "An AIS based feature selection method for software fault prediction." Intelligent Systems (ICIS), 2014 Iranian Conference on. IEEE, 2014.

[8] Catal, Cagatay, Ugur Sevim, and Banu Diri. "Clustering and metrics thresholds based software fault prediction of unlabeled program modules." Information Technology: New Generations, 2009. ITNG'09. Sixth International Conference on. IEEE, 2009.

[9] Mahajan, Er Rohit, Dr Sunil Kumar Gupta, and Rajeev Kumar Bedi. "Comparison of Various Approaches of Software Fault Prediction: A Review."International Journal of Advanced Technology & Engineering Research (IJATER) (2014).

[10] Kaur, Simranjit, Manish Mahajan, and Dr Parvinder S. Sandhu. "Identification of Fault Prone Modules in Open Source Software Systems using Hierarchical based Clustering." ISEMS, Bangkok, July (2011).

[11] Pushpavathi, T. P., V. Suma, and V. Ramaswamy. "Analysis of Software Fault and Defect Prediction by Fuzzy C-Means Clustering and Adaptive Neuro Fuzzy C-Means Clustering."

[12] Kaur, Supreet, and Dinesh Kumar. "Software Fault prediction in object oriented software systems using density based clustering

approach." international journal of research in engineering and technology (IJRET) Vol 1 (2012).
[13] BISI, MANJUBALA, and NEERAJ KUMAR GOYAL. "Early Prediction of Software Fault-Prone Module using Artificial Neural Network." International Journal of Performability Engineering 11.1 (2015)
[14] Zafar, Muhammad Husnain, and Muhammad Ilyas. "A Clustering Based Study of Classification Algorithms." *International Journal of Database Theory and Application* 8.1 (2015): 11-22.
[15] Hall, Tracy, and David sBowes. "The state of machine learning methodology in software fault prediction." Machine Learning and Applications (ICMLA), 2012 11th International Conference on. Vol. 2. IEEE, 2012.